Abstract of the Disclosure

The invention provides a method and system for providing the functionality of dynamically-allocated threads in a multithreaded system, in which the operating system provides only statically-allocated threads. With this functionality, a relatively large number of threads can be maintained without a relatively large amount of overhead (either in memory or processor time,) and it remains possible to produce program code without undue complexity. A plurality of dynamically-allocated threads are simulated using a single statically-allocated thread, but with state information regarding each dynamically-allocated thread maintained within the single statically-allocated thread. The single statically-allocated thread includes, for each procedure call that would otherwise introduce a new simulated thread, a memory block including (1) a relatively small procedure call stack for the new simulated thread, and (2) a relatively small collection of local variables and other state information for the new simulated thread. When using multithreading in the WAFL file system, high concurrency among threads can be maintained without any particular requirement that the program code maintain a substantial amount of state information regarding each dynamically-allocated thread. Each routine in the WAFL file system that expects to be suspended or interrupted need maintain only a collection of entry points into which the routine is re-entered when the suspension or interruption is completed. A feature of the C language preprocessor allows the programmer to generate each of these entry points without substantial additional programming work, with the aid of one or more programming macros.